National College of Ireland

Higher Diploma in Science in Computing (Software Development)

HDSDEV\_SPOL23\_Y2

2024/2025

Thomas Leavy

X23233826

X23233826@student.ncirl.ie

*Developing the Primary Pals e-Learning Platform for Primary School Pupils*

Technical Report



Logo, company name

Description automatically generated

**National College of Ireland**

**Project Submission Sheet – 2024/2025**

Student Name: Thomas Leavy

Student ID: x23233826

Programme: Higher Diploma in Science in Computing (Software Development) (HDSDEV\_SPOL23\_Y2)

Year: 2025

Module: Project

Lecturer: Hamilton Niculescu

Submission Due Date: 26/04/2025

Project Title: Developing the Primary Pals e-Learning Platform for Primary School Pupils

Word Count (excluding bibliography and appendices):

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

|  |  |
| --- | --- |
| **Signature: Thomas Leavy** |  |
| **Date: 14/04/2025** |  |

**Table of Contents**

[Table of Figures 6](#_Toc195477615)

[Glossary, Acronyms, Abbreviations and Definitions 7](#_Toc195477616)

[Executive Summary 8](#_Toc195477617)

[1 Introduction 9](#_Toc195477618)

[1.1 Background 9](#_Toc195477619)

[1.2 Aims 10](#_Toc195477620)

[1.3 Technologies 11](#_Toc195477621)

[1.4 Structure 12](#_Toc195477622)

[2 System 14](#_Toc195477623)

[2.1 Requirements 14](#_Toc195477624)

[2.1.1 Functional requirements 14](#_Toc195477625)

[2.1.2 Data requirements 17](#_Toc195477626)

[2.1.3 User requirements 20](#_Toc195477627)

[2.1.4 Environmental requirements 22](#_Toc195477628)

[2.1.5 Usability requirements 24](#_Toc195477629)

[2.2 Design and Architecture 26](#_Toc195477630)

[2.3 Implementation 28](#_Toc195477631)

[2.4 Testing 30](#_Toc195477632)

[2.5 Graphical User Interface (GUI) Layout 31](#_Toc195477633)

[2.6 Customer testing 35](#_Toc195477634)

[2.7 Evaluation 36](#_Toc195477635)

[3 Conclusions 37](#_Toc195477636)

[4 Further development or research 39](#_Toc195477637)

[5 References 42](#_Toc195477638)

[6 Appendix 44](#_Toc195477639)

[6.1 Project Proposal 44](#_Toc195477640)

[6.2 Project Plan 44](#_Toc195477641)

[6.3 Requirement Specification 44](#_Toc195477642)

[6.4 Monthly Journal 45](#_Toc195477643)

# Table of Figures

|  |  |  |
| --- | --- | --- |
| ***Figure*** | ***Description*** | ***Page Number*** |
| 1 | Class diagram of functional requirements | 16 |
| 2 | ER Diagram demonstrating PostgreSQL database tables and relationships. | 20 |
| 3 | Flow diagram showing various pages of the primary Pals e-Learning Platform | 22 |
| 4 | Technologies used in environmental development | 24 |
| 5 | Welcome page for the Primary Pals e-Learning Platform | 32 |
| 6 | Teacher Dashboard page for the Primary Pals e-Learning Platform | 33 |
| 7 | Pupil Dashboard page for the Primary Pals e-Learning Platform | 34 |
| 8 | Create Quiz page for the Primary Pals e-Learning Platform | 34 |
| 9 | Statistics page for the Primary Pals e-Learning Platform | 35 |

# Glossary, Acronyms, Abbreviations and Definitions

A table containing all the terms and acronyms used in the document.

|  |  |
| --- | --- |
| ***Abbreviation*** | ***Meaning*** |
| API | Application Programming Interface |
| AWS | Amazon Web Services |
| CRUD | Create Read Update Delete |
| CSS | Cascading Style Sheets |
| HTML | Hypertext Markup Language |
| HTTP | Hypertext Transfer Protocol |
| ID | Identification |
| JPA | Java Persistence API |
| JSON | JavaScript Object Notation |
| JWT | JSON Web Token |
| OS | Operating System |
| RBAC | Role-Based Access Control |
| REST | REpresentational State Transfer |
| SQL | Structured Query Language |
| UI/UX | User Interface/Experience |
| VSC | Visual Studio Code |

# Executive Summary

The Primary Pals e-Learning Platform addresses the need for interactive and engaging digital tools in primary schools. Traditional methodologies often struggle with keeping pupils motivated and engaged, while teachers also need efficient ways to manage and track the progress of their pupils. This project offers a full-stack solution to those challenges because it delivers real-time quizzes and feedback for pupils, as well as robust administrative tools for teachers.

This project was developed with Java Spring Boot at the backend and React at the frontend. Primary Pals uses RESTful APIs and a PostgreSQL database for storage of data. A role-based access control system is used to ensure pupils and teachers may only access features relevant to them. Moreover, BCrypt is used to hash passwords and user credentials. Pupils have access to an intuitive and engaging user interface. It supports customisable avatars, quick quiz feedback, and a responsive layout that is usable on desktop, tablet and smartphone. Similarly, teachers can create and assign quizzes, manage pupil associations, and check statistics from previously assigned quizzes.

Evaluation occurred through unit testing, integration testing, and simulated user situations that demonstrated the platform’s performance, reliability and user-oriented design. The load times are efficient, and real-time feedback is effective in cultivating pupil engagement. The Primary Pals e-Learning Platform meets key objectives – however, future improvements could occur, including the addition of advanced security mechanisms, an extension of gamification elements, and full cloud deployment for it to reach its full potential while leaving a lasting educational impact.

# Introduction

## Background

Research shows that the positive impact of technology on education has never been clearer (European Commission, 2020; Haleem et al., 2022; UNESCO, 2023). As a primary school teacher, I have first-hand experience with the challenges that teachers and pupils face in classroom environments. This provides me with a unique perspective for what is needed in an interactive learning platform at primary school level. Play is a crucial aspect of pupil development at primary school level (Department of Education, 2023). Throughout my teaching career, I have observed a set of challenges that encouraged me to undertake this project. For example, maintaining engagement is difficult, and research shows that providing engaging and intuitive resources assists in learning environments (Department of Education, 2023). Similarly, current tools are not tailored towards particular pupils of teachers. This project – ***Primary Pals e-Learning Platform*** – is designed from the viewpoint of a teacher who knows what motivates pupils and grasps how technology enriches learning. By considering important aspects like gamification, personalisation and progress tracking, this innovative platform is teacher- and pupil-friendly while also improving the learning experience.

Traditional methods being used in classrooms are being replaced by digital learning environments (OECD, 2020). Many interactive and personalised e-learning platforms are showing promising results in improving pupil engagement in classrooms and for facilitating differentiated instruction, particularly at the primary school level. This platform is designed to provide an engaging and interactive learning environment for pupils at primary school level. Not only shall it support classroom instruction, but it will also facilitate remote and supplementary learning for pupils. This is crucial for addressing the evolving educational needs that exist in today’s technology-oriented society (OECD, 2020; UNESCO, 2023).

The Primary Pals e-Learning Platform addresses key issues in education. There is a lack of engaging and interactive digital tools for primary school pupils (Hsin et al., 2014). There is a need for clear communication between teachers and pupils, with features that track progress of pupils and their performance. Similarly, there are challenges in designing sustainable systems that cater to the teacher’s administrative needs alongside the learning experiences of primary school pupils.

## Aims

Main Aim:

The main aim of this project has been to develop a full-stack e-learning platform that allows primary school pupils to learn interactively, while also providing teachers with tools to create quizzes, manage pupil associations and monitor their progress.

Secondary Aims:

To provide a user-friendly interface that is engaging and accessible for pupils and teachers.

To have functionality for real-time quiz taking and for tracking statistics while facilitating immediate feedback regarding performance.

To have a system in place that is flexible and scalable while also enabling future enhancements, for example, additional gamification features and data-gathering features for teachers)

Expected Outcomes:

The Primary Pals e-Learning Platform acts as a crucial classroom management tool because of its quiz distribution, and it makes learning more exciting and dynamic for pupils.

As a result, pupil engagement will be enhanced because their motivation will increase since interactive elements are included.

The Primary Pals e-Learning Platform will enable teachers to make data-driven decisions regarding pupils in their classrooms because the platform gathers real-time statistics and tracks pupil progress.

## Technologies

Backend

Java is being used because of its versatility and because it is one of the highest performing virtual machines available (Savitch, 2018; Urma et al., 2015).

Spring Boot (Java) was selected because it provides a robust RESTful API that effectively handles business logic, quizzes, user management and statistics (Walls, 2016). The Spring Data JPA is also being used because it facilitates persistence in terms of data, and it simplifies the connection with the PostgreSQL database.

PostgreSQL was carefully chosen as the relational database because it has powerful planning algorithm that does not require one to optimise their query when using it (Obe and Hsu, 2015). This makes it particularly useful at maintaining tables for users, quizzes, live quiz information, quiz results and individual user statistics.

Frontend

React.js was chosen because it enables developers to build dynamic and component-oriented user interfaces that are interactive and responsive in design (Banks, 2017). The React Router was used to manage the client-side routing, while also enabling efficient navigation between different sections of the application. Axios was also selected because it handles the HTTP requests between the React frontend and the Spring Boot backend.

Deployment and Build Tools

Maven was chosen in this project because it is excellent for managing dependencies and for building the Spring Boot application.

Npm create-react-app is an essential tool that is used because it manages the frontend project dependencies effectively and is useful for building the React app in a “ready for production” format (Banks, 2017).

Git and GitHub were used to facilitate the repository and for version control to effectively manage the codebase.

pgAdmin 4 was chosen to query, modify and manage the PostgreSQL relational database being used for the Primary Pals e-Learning Platform.

Further tools used

Visual Studio Code was used as the primary code development environment. Similarly, the Windows Command Prompt was used to execute Maven and npm commands.

## Structure

This technical report is organised in the following format:

Chapter 1 introduces the project with important background information, aims and objectives, as well as the main technologies used, and the rationale for developing an interactive e-learning platform for primary school pupils. In essence, it sets the stage for the descriptions outlined in later chapter.

Chapter 2 presents an important discussion of the platform’s design and its implementation. This chapter covers essential requirements (for example, functional, data, user, environmental, and usability), while underscoring how the platform is useful for both teacher and pupil users respectively. Following this, the chapter outlines the overall system architecture, design decisions that were implemented, and the collaboration between the frontend and backend components. Similarly, Chapter 2 details the strategies used for implementation in terms of layers, the testing approach that used automated unit and integration tests, and the measures that were taken to ensure total data integrity and performance.

Chapter 3 summarises the outcomes of the project through the evaluation of its strengths and weaknesses. It discusses challenges that had been encountered throughout development, and highlights opportunities for future development and enhancement.

Chapter 4 offers a crucial reflection on the potential improvements and further research directions. For example, it considers additional resources, including cloud deployment, advanced security, greater gamification features, richer analytic capabilities, and improved user management. Chapter 4 also indicates the scope for transforming the platform into a more robust and scalable educational tool.

Chapter 5 lists all references that were referenced in the technical report, and this follows the Harvard referencing system.

Chapter 6 outlines additional documents that supported the development of the project, for example, the project proposal, the project plan, the requirements specifications and the use of a monthly journal. Each of these elements aided in establishing clear objectives, clarifying requirements, and guiding the direction of the project.

# System

## Requirements

During the course of the development of this project, the original requirements specification changed. This was necessary because the usability, functionality and scalability required greater refinement. The below sections outline the current requirements:

### Functional requirements

* User management – registering and logging in

It was vital to ensure pupils and teachers could create individual accounts with separate features. Similarly, it was necessary that a user’s credentials would be validated, for example, ensuring an @ symbol was included in the email address. It was also important that a user’s password was stored securely using hashing mechanisms, for example, BCrypt.

* User management – teacher/pupil associations

Individual teacher users needed to be able to view and manage pupils associated with them. Pupils also needed to have permissions to change their teacher to the one they were associated with. It was essential that pupils and teachers had associations because otherwise, pupils would have access to all quizzes created by every teacher using the Primary Pals e-Learning Platform.

* Quiz Management – quiz creating and assigning quiz

It was necessary that teacher users were able to create quizzes with up to 5 questions, 4 choices, and to have a variety of answering options available. Similarly, once the teacher users created their quiz, they needed to receive a uniquely generated code for each quiz created that would be used for pupils to access the quiz.

* Quiz Management – live quiz participation

It was necessary that pupil users were able to participate in a quiz via the quiz code that had been automatically generated for the specific quiz. Upon completing the quiz, it was essential that the pupil user’s results would be captured and recorded in the PostgreSQL database. The correctness of each response from the pupil user would be stored in real time.

* Statistics and Feedback – real-time feedback

It was important that the quiz statistics would be updated and displayed after the quiz had been submitted by the pupil user. Moreover, the amount of correct answers submitted needed to be calculated and each pupil user’s performance had to be recorded.

* Statistics and Feedback – results viewing

A page needed to be provided for teachers to view and access quiz results/statistics upon completion by their associated pupil users. With this gathered data, it would then be possible for pupil users to receive a detailed breakdown of their quiz results, including whether specific answers were correct/incorrect, as well as their overall score in that particular quiz.

* Avatar selection

This gamification feature was included because it enabled pupils to personalise their account and to use accumulated points from completed quizzes to unlock new avatars for their accounts. In essence, this was a motivation feature that would ensure pupil users returned to the e-learning platform to complete more quizzes and unlock more avatar choices for their profile. It was decided that the selected avatar would be displayed across the platform as part of the profile personalisation element of the Primary Pals e-Learning Platform.

* Integration and Inter-connectedness

For the Primary Pals e-Learning Platform, it was essential that there was efficient communication between the React frontend and the Java Spring Boot backend with the use of RESTful APIs. Furthermore, it was necessary that data was able to flow from the user interface to the database with Spring Data JPA and PostgreSQL, and that this occurred in a reliable manner.

* Data Integrity and Security Elements

For this project, data integrity was a core concern that needed to be addressed – and to ensure its presence across user, quiz and statistics records, a normalised database design was chosen. Similarly, RBAC (role-based access control) was utilised to guarantee that teacher users and pupil users could only access respective functionalities as per their accounts.

*Figure 1 – Class Diagram made with Mermaid Live Editor*

A screenshot of a computer

AI-generated content may be incorrect.

### Data requirements

Database schemas were requirements in developing the Primary Pals e-Learning Platform:

User Table

Purpose: This was needed to store profiles for pupil users and teacher users.

Main Fields:

* User ID: This was a primary key for each unique user.
* Email: This included the user’s email address and needed to be unique.
* Username: This would act as the user’s displayed name.
* Hashed Passwords: It was necessary that a user’s password was stored in the database with cryptographic hashing to ensure GDPR compliance and for complying with privacy standards. Similarly, this would significantly decrease the likelihood of unauthorised access to user accounts.
* Role: This was important for indicating whether the user was a teacher or a pupil.
* Associated Teacher: In terms of pupil records, it was essential that they associated with a teacher using the teacher’s unique user ID (this was a foreign key for the Users table).

Quizzes and Questions Tables

Purpose: This was important because it stored metadata about each quiz created.

Main Fields:

* Quiz ID: This acted as the unique identifier for each quiz created.
* Title: This acted as the title for each created quiz.
* Code: A unique code was generated for each quiz created, and pupils entered this code to access the quiz.
* Teacher Association: This displayed the teacher user ID for the particular quiz that had been created.
* Creation Date: This was a timestamp that marked when the quiz was made.

Questions Table

Purpose: This was essential because it ensured each question in a quiz was stored separately in the database while also being linked to its parent quiz.

Main Fields:

* Question ID: This acted as a unique identifier for the question.
* Quiz ID: This was a foreign key that tied in with the associated parent quiz.
* Question Text: This was the prompt that was given to the pupil users when participating in the quiz.
* Answer Choices: This included 4 fields for storing potential answers (for example, statement 1, statement 2, statement 3, statement 4).
* Correct Answer: This acted as the identifier of the correct answer, for example, in one question, it could be statement 2, and in another question, it could be statement 3.

Quiz Live Results Table

Purpose: This table was necessary because it recorded the real-time quiz submissions of the pupil users.

Main Fields:

* Result ID: This acted as the unique identifier for each quiz attempt.
* User ID: This referenced the pupil user who submitted an answer.
* Quiz ID and Question Number: This involved identifying which quiz and which question the particular response belonged to.
* Answer: This acted as the pupil’s submitted answer.
* Correct: This was a Boolean value that indicated if the answer that was submitted was correct/incorrect.
* Timestamp: This showcased when the answer was submitted.
* Submission ID: This was a unique identifier that grouped responses from the same particular quiz submission.

Statistics Table

Purpose: This table was important for maintaining accumulated records of pupil performance and their usage metrics of the Primary Pals e-Learning Platform.

Main Fields:

* Statistics ID: This acted as the unique identifier for the statistics record.
* User ID: This showcased the pupil user’s unique identifier.
* Quizzes Completed: This counted the amount of quizzes completed by the pupil user.
* Points Earned: This included the total points accumulated by the pupil user (for example, points were awarded on a per-correct answer basis).
* Session Duration: This recorded the login times and the cumulative session durations. This count was kept, even if the pupil user logged out.

Data Integrity

Referential Integrity – Foreign Keys

* It was necessary to ensure certain fields, for example, teacherId , quizId, and userId referenced the primary key within their parent tables respectively. To maintain data consistency, it was essential to enforce NOT NULL and particular constraints on main fields like email, quiz code, etc.

Normalisation

* The database has been designed to separate data into various logic-oriented tables (for example, Users, Quizzes, Quiz Questions, Quiz Live, and Statistics), This occurred to avoid redundancy and to maintain consistency.

Indexes

* Indexes were implemented on columns that were often queried (for example, quiz code, user ID), in an effort to improve query performance.

Backup and Recovery

* It was important to have regular backups through tools like pgAdmin 4, to ensure data would be restored in the event of data loss.

*Figure 2 – ER Diagram demonstrating PostgreSQL database tables and relationships.*

A screenshot of a computer

AI-generated content may be incorrect.

### User requirements

For pupil users, it was necessary that there was a simple but intuitive interface, to encourage engaging learning experiences (Department of Education, 2023). This needed to be tailored for young learners and a focus on ease-of-use was imperative. Consequently, large buttons, clear labels and various visual cues were necessary. Completing quizzes would also result in pupils being able to see their results immediately. It was also required that gamification elements existed for pupils – therefore, features existed that enabled them to choose and unlock avatars that would be visible across the platform on their profile, to see how many quizzes they completed, their accumulated login time, and the number of points they had accumulated from correct answers. Pupils also needed to be able to disconnect from one associated teacher, and to re-associate with a new teacher.

For teacher users, it was essential that there were comprehensive quiz management tools available, A user-friendly interface was central to the project design, particularly regarding elements like creating, editing and assigning quizzes to pupil users. There needed to be a feature that generated unique quiz codes so secure distribution of the quiz could occur. Similarly, structured forms with guided and balanced workflows were necessary to simplify the quiz making process, especially to provide ease-of-access. It was also vital that teachers could view pupil user results from completed quizzes – this would show which answers pupils got correct, and the average score among all pupils who participated in a particular quiz. Similar to the pupil user, it was important that the teacher could disconnect pupils with whom they were no longer associated, and to reconnect pupil users who were new associations (for example, new pupils in their class).

It was necessary that the Primary Pals e-Learning Platform was built with a responsive design in mind. For example, if a user accessed the platform from a tablet device, the interface needed to adapt automatically to this screen size. This also included touch-friendly elements that facilitated ease of interaction on mobile devices.

*Figure 3 – Flow diagram showing various pages of the primary Pals e-Learning Platform*

A screenshot of a computer

AI-generated content may be incorrect.

The Primary Pals e-Learning Platform ensures user data is protected, and access is managed in a secure manner. User authentication uses BCrypt for password hashing to ensure passwords are stored securely. RBAC restricts certain functionalities as per the particular account (for example, pupil account or teacher account). Therefore, teacher users have administrative privileges unavailable to pupil users. Although advanced token-based authentication (JWT) was considered, ultimately it was not included in the project because the current scope satisfies the requirements of the project.

### Environmental requirements

Development Environment – operating system/tools

* The main development of this project occurred on a Windows OS with Visual Studio Code as the integrated development environment. This was also combined with the Windows Command Prompt and the integrated terminal in VS Code, to complete commands.

Development Environment – Dependency Management

* Maven was used for backend development (for example, Spring Boot), to handle packaging, dependencies and building of the project. Similarly, npm with create-react-app was used to manage frontend dependencies, local development on the operating system, and scripts used to build the React-based application.

Development Environment – Local Database Management

* PostgreSQL was installed locally during the development of the Primary Pals e-Learning Platform. pgAdmin 4 was used to query and manage the database.

Production Environment

* The project is hosted on GitHub in a repository. This repository has all production-ready code for the backend/frontend, and all related configuration/build scripts. Moreover, the repository contains a copy of the PostgreSQL database dump. This was necessary because it demonstrates that the whole system can be recreated, ensuring transparency of the project. It is available at this link: <https://github.com/thomasleavy/eLearningApplication>.

Although cloud deployment through AWS/Heroku was considered, hosting it through GitHub was sufficient for demonstrating the full codebase and its documentation.

Testing Environment – Local Testing

* Backend testing occurred using JUnit and Spring Boot Test for integration and unit testing. These were features that came with the Maven configuration. Similarly, frontend testing occurred with React Testing Library and Jest to make sure all components appeared as expected. These features came with Create React App.

Testing Environment – database testing

* pgAdmin 4 was necessary during development to test the database directly - for example, by running SQL queries to verify the data integrity, checking table content/relationships and ensuring changes to the database schema occurred efficiently and without issue.

*Figure 4 – Technologies used in environmental development*

A screenshot of a computer

AI-generated content may be incorrect.

### Usability requirements

User Interface – intuitive design

* The user interface was designed with user-friendliness as a central goal. This was done because it was important both pupil and teacher users could navigate the Primary Pals e-Learning Platform without issues. Furthermore, visual elements (for example, colours, icons, etc.) were chosen carefully to assist with comprehension and to improve the overall user experience.

User Interface – consistency

* It was necessary that the e-Learning Platform had consistency across all pages. For example, having the same design elements for design standardisation, similar colouring, having the Primary Pals header across all pages, having a footer with relevant links, etc. was vital for this project.

Accessibility and Responsive Design – responsive layout

* The design of the Primary Pals e-Learning Platform adapts efficiently to various screen sizes, for example, desktops, tablets and smartphones. This ensures users of various device screen sizes have a consistent experience of the platform, irrespective of screen size. Moreover, interface components are touch-friendly with suitably sized buttons and links for mobile users.

Accessibility and Responsive Design – Navigation and Considerations

* Menus and buttons are visible clearly for users, allowing them to navigate forward and backward through pages without confusion. The user also receives immediate visual feedback to help guide them through the e-Learning Platform. For example, this includes button hover effects and error messages. The user interface also has high contrast text, alternative text for images to improve accessibility and inclusion of all users, and semantic HTML for supporting users that may have disabilities.

Performance – load times

* It was important to ensure the front-end assts (for example, images, JavaScript, CSS) were optimised to have fast and efficient loading times. Similarly, the architecture and server-side endpoints of the Primary Pals e-Learning Platform were designed for well-planned processing of data.

Performance – real-time feedback

* It was necessary that participation in and completion of quizzes would result in immediate feedback for the pupil user – this would then assist the pupil user in understanding their progress. Likewise, live statistics needed to be updated in a timely manner because teacher users rely on data that is up-to-date for monitoring the progress of their assigned pupils. These statistics would then help the teacher to identify trends/challenges.

## Design and Architecture

The Primary Pals e-Learning Platform has various architectural elements that leverage modern web development frameworks in order to separate concerns and to enhance the upkeep of the overall platform. For example:

Overall Architecture – presentation layer

* In terms of the frontend development, this was build using React, because it provides an interactive user interface and a responsive design that is tailorable to pupil and teacher users respectively. The React Router is also excellent for handling client-side navigation, while Axios is used for facilitating HTTP communication.

Overall Architecture – application layer

* In terms of the backend development, Java Spring Boot was used because it exposes RESTful APIs to authenticate users, for management of quizzes, for statistics aggregation and for managing pupil-teacher associations. All business logic is encapsulated in service (for example, QuizService.java, QuizLiveService.java, StatisticsService.java, and UserService.java), while Spring Data JPA is used to manage database interactions.

Overall Architecture – data layer

* Persistence was required because it provides a robust mechanism that can be used to retrieve, store and manage the data that is essential for the Primary Pals e-Learning Platform. For example, by using a normalised PostgreSQL database, data like user profiles, quiz content, live quiz responses and statistics may be maintained with little redundancy and high integrity.

Overall Architecture – deployment environment

* The Primary Pals e-Learning Platform is hosted as a complete codebase in a repository on GitHub. It includes a copy of the database dump file: <https://github.com/thomasleavy/eLearningApplication>. Cloud-based hosting was considered, but ultimately not included because the GitHub repository represents the production-ready platform.

Main Algorithms – quiz code generation

* To create a unique quiz code, a random string is created. In terms of mathematical notation, 𝒞 in this case represents the set of characters allowed. A pin code of length n is then created by choosing characters at random from 𝒞. For example:

Main Algorithms – data validation and security

* All user inputs are validated with regular expressions and business rules (for example, emails must contain an @ symbol) to cater for data consistency. Passwords are also hashed with BCrypt. For example, if P denotes the plain-test password, then the stored password becomes:

Therefore, this ensures that if there is unauthorised access to the database, the original passwords are unable to be easily recovered.

## Implementation

The Primary Pals e-Learning Platform is implemented with a two-layer approach – it uses a Spring Boot backend and a React frontend. Both layers communicate using APIs with JSON.

Backend Implementation – controllers

* Controllers like QuizController.java, UserController.java, PupilController.java and StatisticsController.java expose REST endpoints for main functionalities. For example, the QuizController.java file handles quiz creation, live submissions and result aggregation. Below is an excerpt of code that demonstrates how the platform processes live quiz submissions – the logic shown aggregates the amount of correct responses, and updates the user’s points appropriately:

int correctCount = (int) quizLives.stream()

                .filter(ql -> Boolean.TRUE.equals(ql.getCorrect()))

                .count();

        statisticsService.addPoints(userId, correctCount);

Backend Implementation – services

* The main business logic is encapsulated in services like QuizService.java, QuizLiveService.java and StatisticsService.java. These manage interactions with the PostgreSQL database through the Spring Data JPA.

Backend Implementation – repositories

* The repository interfaces facilitate create, read, update and delete (CRUD) operations and enforces data integrity with normalised schema design.

Frontend Implementations – components

* React components like WelcomePage.js, TeacherDashboard.js, PupilDashboard.js, Quiz.js and SelectAvatar.js have a dynamic and responsive user interface. The below code excerpt from Quiz.js shows this rendering of quiz questions and a submit button that triggers an API call:

return (

    <div className="quiz">

      <PupilHeader />

      <h1>{quizData.quizTitle}</h1>

      {quizData.questions.map((q, qIndex) => (

        <div key={qIndex} className="question-block">

          <h3>{q.question}</h3>

          <ul>

            <li>

              <label>

                <input

                  type="radio"

                  name={`question-${qIndex}`}

                  value="0"

                  checked={answers[qIndex] === 0}

                  onChange={() => handleOptionChange(qIndex, 0)}

                />

                {q.statement1}

              </label>

            </li>

            <li>

              <label>

                <input

                  type="radio"

                  name={`question-${qIndex}`}

                  value="1"

                  checked={answers[qIndex] === 1}

                  onChange={() => handleOptionChange(qIndex, 1)}

                />

                {q.statement2}

              </label>

            </li>

            <li>

              <label>

                <input

                  type="radio"

                  name={`question-${qIndex}`}

                  value="2"

                  checked={answers[qIndex] === 2}

                  onChange={() => handleOptionChange(qIndex, 2)}

                />

                {q.statement3}

              </label>

            </li>

            <li>

              <label>

                <input

                  type="radio"

                  name={`question-${qIndex}`}

                  value="3"

                  checked={answers[qIndex] === 3}

                  onChange={() => handleOptionChange(qIndex, 3)}

                />

                {q.statement4}

              </label>

            </li>

          </ul>

        </div>

      ))}

      <button className="submit-button" onClick={handleSubmitQuiz}>Submit Quiz</button>

    </div>

  );

Frontend Implementation – integration

* Axios is utilised for the frontend to interact with the endpoints of the backend, and the React Router is used to manage navigation across different pages.

## Testing

Backend testing – Junit and Spring Boot Test

* Unit tests in QuizLiveServiceTest.java use Junit and Mockito to validate individual methods, for example:

QuizLiveService.saveAllQuizLives()

Similarly, integration tests in the ElearnApplicationTests.java class are configured to load Spring context and verify the relations between the repositories, services and controllers.

Backend testing – test cases

* These tests cover endpoints, for example, login, quiz submission, and statistics updates. They give provisions for simulating edge cases, for example, as seen in testing scenarios and error handling situations.

Frontend Testing – React testing library and Jest

* The Primary Pals e-Learning Platform was built with Create React App. This comes with Jest and the React Testing Library. With this configuration, testing infrastructure is provided to verify component rendering, validity of forms, and user navigation.

Database Testing – pgAdmin 4

* pgAdmin 4 has been used to run SQL queries and to confirm the integrity of various schema. In turn, this has ensured that CRUD operations are reflected correctly in the PostgreSQL database.

Continuous Integration

* Tests are automated in the build via maven for the backend and Create React App for the frontend. This makes sure code changes are validated regularly and are unable to break the current functionality.

## Graphical User Interface (GUI) Layout

The Primary Pals e-Learning Platform have a clean, intuitive and responsive user interface that is tailored for primary school pupils and teachers. Below are sample screenshots with descriptions:

Welcome Page

* This page showcases the initial landing features with a header and footer section. It features two distinct containers for registering with the platform and for logging into the platform. Each has large input fields with streamlined forms that are designed with ease-of-access in mind for both young learners and teachers:

*Figure 5 – Welcome page for the Primary Pals e-Learning Platform*

***A screenshot of a computer

AI-generated content may be incorrect.***

Teacher Dashboard

* This page represents the main interface for managing quizzes, viewing pupil results, and handling pupil associations. It is the homepage for teacher users. Important buttons, like Create/Assign Quiz and Pupil Results, provide quick navigation to main features while consistent colouring and layouts reinforces the standard design of the entire e-learning platform:

*Figure 6 – Teacher Dashboard page for the Primary Pals e-Learning Platform*

***A screenshot of a computer

AI-generated content may be incorrect.***

Pupil Dashboard

* This is the main homepage for pupil users of the Primary Pals e-Learning Platform. It enables pupil to access quizzes, get real-time feedback, associate with a teacher, and examine their account statistics. It is an intuitive and easily accessible layout that combines personalised elements like a pupil’s chosen avatar, which motivates pupils to continue earning points from completing quizzes while fostering positive learning experiences.

*Figure 7 – Pupil Dashboard page for the Primary Pals e-Learning Platform*

***A screenshot of a computer

AI-generated content may be incorrect.***

Create Quiz Page

* This page enables teacher users to create, edit and assign quizzes easily. It contains a well-structured form with 5 questions per quiz and 4 possible answers. Teacher users can input a quiz title while also defining correct answers for each question. The teacher can also update previous quizzes and there is real-time validation and user-friendly error messages that help to ensure data accuracy prior to creating a quiz.

*Figure 8 – Create Quiz page for the Primary Pals e-Learning Platform*

***A screenshot of a computer

AI-generated content may be incorrect.***

Statistics Page

* This page provides pupils with the opportunity to see statistics from their account. For example, the pupil user can see how many points they have earned, their accumulated login time, and how many quizzes they have completed.

*Figure 9 – Statistics page for the Primary Pals e-Learning Platform*

A screenshot of a computer

AI-generated content may be incorrect.

## Customer testing

Due to time and resource constraints, customer testing was not conducted for this project. Instead, simulated user tests occurred with representative test cases and reviews. Simulations had typical user scenarios for pupil and teacher users, for example, to verify that the platform met the key requirements. Simulated tests examined important functionalities, for example, account registration, quiz participation and statistical updates like timing the length in which a user had been logged in. Although this approach does not replace the insights and experience of real-life users, it provides valuable insights regarding the functionality and user experience of the Primary Pals e-Learning Platform.

## Evaluation

The Primary Pals e-Learning Platform was evaluated with a combination of automated tests (for example, unit, UI and integration tests), and simulated user situations. Quantitative metrics were acquired from automated testing, for example, with quiz submissions and result aggregation, and qualitative assessments were achieved from simulated interactions. The performance evaluations showcased fast and efficient load times with immediate feedback from completing quizzes. Moreover, scalability was upheld through a normalised PostgreSQL database with an efficient API design. The responsive user interface was examined across desktop and mobile environments with simulation tools.

Overall, the evaluation shows that the Primary Pals e-Learning Platform meets particular design necessities regarding functionality, performance and usability, with continuous integration that ensures new updates cannot compromise existing features.

# Conclusions

The Primary Pals e-Learning Platform fulfils its core objectives with delivering an interactive and intuitive environment for primary school pupils. It also provides teachers with necessary tools to create quizzes, manage pupils, and track performance of their assigned pupils. During the development, various challenges emerged – for example, time management, choosing the most suitable libraries and plugins, and ensuring the UI/UX designs were engaging aesthetically whilst also accessible to younger users.

Advantage – clear functionality

* The Primary Pals e-Learning Platform offers a full-stack solution, built with Spring Boot in the backend, and React in the frontend. It therefore facilitates real-time quiz participation, live statistics, and automated feedback for users.

Advantage – user-friendly design

* The platform prioritises intuitive navigation while maintaining a responsive layout that is suited to primary school pupils and teachers alike. It also features consistent colouring and layout designs across all pages, thus simplifying workflows while simultaneously enhancing the overall user experience.

Advantage – secure, scalable architecture

* The Primary Pals e-Learning Platform used BCrypt for password hashing in order to safeguard user credentials. Similarly, it uses a normalised PostgreSQL database to ensure robust data integrity, while also supporting any further scalability in future development.

Disadvantage – limited real-world testing

* A disadvantage to this project is that there was no real-life testing of the platform. For example, customer testing relied solely on simulations rather than feedback from real users. This may hinder insights regarding user behaviour.

Disadvantage – deployment constraints

* While the platform is currently showcased through a GitHub repository, it is not accessible through a live cloud environment, which may impact its availability for external users.

Disadvantage – advanced security and optimisation was not implemented

* Features like JWT-oriented authentication and heavier load optimisation were outside the scope of this version of Primary Pals but could be implemented at a future date.

Disadvantage – exclusion of additional features

* Additional gamification features, more game-oriented quiz interfaces, additional unlockable features for pupil users and a more dynamic and animated user interface were a consideration for this project. However, because of time constraints, and since the current version meets prioritised and necessary goals, these were not implemented.

# Further development or research

Upon reflection, while the Primary Pals e-Learning Platform currently meets its primary objectives, several routes for future developments and research exist that could improve its functionality, security, performance and user engagement. With added resources and time, the following enhancements and research routes could potentially be implemented:

Further Development – Cloud deployment and scalability

* Currently, the production version of the e-learning platform is available through a GitHub repository. While this provides full code transparency, it has limited live access. An improvement would involve deploying it on a cloud platform such as AWS, Heroku or Azure. If this occurred, higher loads and auto-scaling opportunities may present themselves, thus ensuring the platform was responsive as the number of real-time users increased. Similarly, cloud deployment would ensure there is better availability globally and tolerance towards faults. Cloud services would also integrate monitoring/logging tools (for example, AWS offers CloudWatch), and this would help with diagnosing issues, tracking metrics in terms of performance, and upkeeping the overall wellbeing of the system.

Further Development – Enhanced Security

* While the current implementation uses BCrypt for password hashing in order to safeguard information passed through by users, upon reflection, additional improvements to security could be implemented. F or example, using token-based authentication, like JWT, would facilitate stateless session management while also providing more robust protection for API endpoints. Furthermore, in a live production environment, using HTTPS through Spring Boot SSL would make sure data transferral is secure when send over the internet. Similarly, maintaining regular security audits with automated penetration tests would be useful features to integrate into the continuous integration pipeline. This would be crucial for detecting and addressing any potential vulnerabilities in the system.

Further Development – gamification and user engagement features

* User engagement is an essential aspect of the Primary Pals e-Learning Platform, and the current gamification features (for example, avatar selection) can be expanded u[on to enhance motivation and continuous usage of the platform. For example, introducing a greater selection of avatars for various milestones that can be examined in the pupil statistics page, and introducing leaderboards to compare performance among peers could incentivise pupil users to use the platform further. Furthermore, utilising interactive animations during transitions or after completing a quiz successfully would provide further positive reinforcement for pupil users to engage with the platform. A future version may also include a more refined reward system where points unlock more interactive content and customisation features.

Further Development – greater reporting of analytics

* Currently, the e-learning platform aggregates essential performance metrics, however, there is potential to gather a richer and greater amount of data. For example, collecting data on user interactions, like how long a user spent on a question or examining some common mistake patterns in terms of answers, would offer valuable insights into the learning behaviour of pupil users. Teachers would be able to use this data to identify difficult learning topics and adjust instructional strategies as a result. Similarly, enabling teachers to generate customisable reports that visualise various trends over time (for example, with charts/graphs), would improve the system’s value as a tool in the classroom.

Further Development – improved frontend functionality

* The React frontend is excellent at delivering a responsive user interface; however, it may be improved by refactoring various components to improve reusability and modularity to simplify later updates. For example, using a state management solution like Context API would assist in managing complex state interactions more efficiently as the e-learning platform scales. Likewise, converting the platform into a Progressive Web-based Application would enable users to install it onto their device, thus improving the mobile experience and providing offline features. Integrating real-time notifications with WebSockets would alert users about new quizzes or announcements made by teachers. In turn, this would make the e-learning platform more dynamic.

Further Development – user management and role-oriented enhancements

* The Primary Pals e-Learning Platform provides essential user management and role-based access. Nonetheless, upon reflection, future research may explore advanced RBAC in which more nuanced permission levels could facilitate more specific controls in terms of access. This could include different teacher roles (for example, assistant teacher, student teacher, etc.) with varying permissions. Similarly, providing greater personalisation features (for example, being able to customise an account further) and learning pathways based on performance information, would cater more specifically to individual learning needs, thus improving user engagement.

Overall, the Primary Pals e-Learning Platform already offers a robust and interactive learning experience – however, future improvements regarding cloud deployment, security, gamification features, analytics, frontend functionality and refined user management could be introduced. These enhancements would improve the platform’s capabilities and broaden its usage in real-life educational environments.

# References

Banks, A. and Porcello, E. (2017). *Learning React*. [online] Available at: https://phanindra-reddy.github.io/portfolio/static/media/learningreact.da253db3.pdf [Accessed 4 February 2025].

Department of Education. (2023). *Primary Curriculum Framework.* [online] Available at: <https://www.curriculumonline.ie/getmedia/84747851-0581-431b-b4d7-dc6ee850883e/2023-Primary-Framework-ENG-screen.pdf> [Accessed 13 April 2025].

European Commission. (2020). *Digital Education Plan 2021-2027*. [online] Available at: <https://education.ec.europa.eu/sites/default/files/document-library-docs/deap-communication-sept2020_en.pdf> [Accessed 13 April 2025].

Haleem, A., Javaid, M., Asim Qadri, M. and Suman, R. (2022). ‘Understanding the role of digital technologies in education: A review’, *ScienceDirect*. [online] Available at: <https://doi.org/10.1016/j.susoc.2022.05.004>.

Hsin, W.J., Li, M.C. and Tsai, C.C. (2014). ‘The influence of young children’s use of technology on their learning: A review of the literature’, *Educational Technology & Society*, 17(4), pp. 85–99. Available at: <https://www.jstor.org/stable/jeductechsoci.17.4.85>. [Accessed 13April 2025].

Obe, R. and Hsu, L. (2015). *PostgreSQL: Up and Running*. [online] Available at: https://k0d.cc/storage/books/Databases/Postgresql/PostgreSQL%20Up%20and%20Running,%202nd%20Edition.pdf [Accessed 4 February 2025].

OECD. (2020). *Education at a Glance 2020 OECD Indicators*. [online] Available at: <https://www.oecd.org/content/dam/oecd/en/publications/reports/2020/09/education-at-a-glance-2020_19b01e87/69096873-en.pdf> [Accessed 11 April 2025].

Savitch, W. (2018). *Java An Introduction to Problem Solving Programming*, 8th ed., Pearson: London.

UNESCO. (2023). *Technology in education: A tool on whose terms*?. [online] Available at: <https://www.unesco.org/gem-report/sites/default/files/medias/fichiers/2023/07/Summary_v5.pdf> [Accessed 12 April 2025].

Urma, R.G., Fusco, M. and Mycroft, A. (2015). *Java 8 in Action*, Manning: Shelter Island, NY.

Walls, C. (2016). *Spring Boot in Action*. [online] Available at: https://www.nitinagrawal.com/uploads/2/1/3/6/21361954/spring\_boot\_in\_action.pdf [Accessed 4 February 2025].

# Appendix

## Project Proposal

The project proposal was submitted to Moodle on 07/02/2025. It included proposals in terms of objectives, background, technical approach, special resources required, project plan, technical details, and an evaluation. It was essential for guiding the initial direction of the project.

## Project Plan

The project plan was guided by various submission deadlines as seen in the table below:

|  |  |  |
| --- | --- | --- |
| Document | Brief Description | Submitted |
| Project Proposal | Outline of project concept | Yes |
| Ethics Approval Form | Ethical considerations | Yes |
| Project Requirement Specification | Describes what application will do and how it is expected to perform | Yes |
| Interim Report | Current status of project | Yes |
| Project Analysis and Design Documentation | System design and architecture of the project | Yes |
| Project Final Report | Facets of completed project | Yes |
| Declaration Cover Sheet | Confirmation of own work | Yes |
| Project Code | Code for the application | Yes |
| Video of Final Project | Presentation video piece | Yes |

## Requirement Specification

The requirement specification was submitted on 22/02/2025. It included vital elements like the purpose of the project, its scope, user requirements, requirement specifications, functional and non-functional requirements. It also enabled a plan of action to be considered in terms of the graphical user interface, system architecture, and system evolution.

## Monthly Journal

A monthly journal was kept in order to track progress. It was used for brainstorming ideas and discerning what was best for being included in the project, and what could be excluded. The journal was useful for recording what needed to occur next, and proposed solutions to problems that had been encountered.